

FILE ID**MACDEF

M 5

MM	MM	MM	AAAAAA	CCCCCCC	DDDDDDDD	EEEEEEEEE	FFFFFFFFF
MM	MM	MM	AAAAAA	CCCCCCC	DDDDDDDD	EEEEEEEEE	FFFFFFFFF
MMMM	MMMM	AA	AA	CC	DD	EE	FF
MMMM	MMMM	AA	AA	CC	DD	EE	FF
MM	MM	MM	AA	CC	DD	EE	FF
MM	MM	MM	AA	CC	DD	EE	FF
MM	MM	AA	AA	CC	DD	EEEEEEEEE	FFFFFFFFF
MM	MM	AA	AA	CC	DD	EEEEEEEEE	FFFFFFFFF
MM	MM	AAAAAAA	CC	DD	DD	EE	FF
MM	MM	AAAAAAA	CC	DD	DD	EE	FF
MM	MM	AA	AA	CC	DD	EE	FF
MM	MM	AA	AA	CC	DD	EE	FF
MM	MM	AA	AA	CC	DD	EE	FF
MM	MM	AA	AA	CCCCCCC	DDDDDDDD	EEEEEEEEE	FF
MM	MM	AA	AA	CCCCCCC	DDDDDDDD	EEEEEEEEE	FF

The diagram illustrates three sets of binary strings, each consisting of 10 strings of length 10:

- Left Set:** All strings start with 'L'.
 - String 1: LLLLLLLLLL
 - String 2: LLLLLLLLLL
 - String 3: LLLLLLLLLL
 - String 4: LLLLLLLLLL
 - String 5: LLLLLLLLLL
 - String 6: LLLLLLLLLL
 - String 7: LLLLLLLLLL
 - String 8: LLLLLLLLLL
 - String 9: LLLLLLLLLL
 - String 10: LLLLLLLLLL
- Middle Set:** All strings end with 'I'.
 - String 1: IIIIIIIIII
 - String 2: IIIIIIIIII
 - String 3: IIIIIIIIII
 - String 4: IIIIIIIIII
 - String 5: IIIIIIIIII
 - String 6: IIIIIIIIII
 - String 7: IIIIIIIIII
 - String 8: IIIIIIIIII
 - String 9: IIIIIIIIII
 - String 10: IIIIIIIIII
- Right Set:** All strings end with 'S'.
 - String 1: SSSSSSSSSS
 - String 2: SSSSSSSSSS
 - String 3: SSSSSSSSSS
 - String 4: SSSSSSSSSS
 - String 5: SSSSSSSSSS
 - String 6: SSSSSSSSSS
 - String 7: SSSSSSSSSS
 - String 8: SSSSSSSSSS
 - String 9: SSSSSSSSSS
 - String 10: SSSSSSSSSS

(2)	62	DECLARATIONS
(3)	86	MACRO DEFINE A MACRO
(4)	154	DELETE .MDELETE DIRECTIVE
(5)	190	FREE UP MACRO DELETE OLD MACRO DEFINITION
(6)	245	MACRO_DEFINE CREATE MNB AND SET IT UP
(7)	284	MACRO_ARG_SCAN SCAN FORMAL MACRO ARGUMENTS

```
0000 1 .TITLE MAC$MACDEF      MACRO DEFINITION PROCESSOR
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 ****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25
0000 26 ****
0000 27
0000 28
0000 29 ++
0000 30 FACILITY:      VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35 modules for input to the VAX-11 LINKER.
0000 36
0000 37 ENVIRONMENT: USER MODE
0000 38
0000 39 AUTHOR: Benn Schreiber, CREATION DATE: 20-AUG-78
0000 40
0000 41 MODIFIED BY:
0000 42
0000 43     V03.00 MTR0009      Mike Rhodes      15-Mar-1982
0000 44             Modify routine MACRO:: to de-optimize the handling of
0000 45             null macros. Fixes SPR #11-42119.
0000 46
0000 47     V01.10 RN0023      R. Newland      3-Nov-1979
0000 48             New message codes to get error messages from system
0000 49             message file.
0000 50
0000 51     V01.09 RN0010      R. Newland      5-Sep-1979
0000 52             Multipage MXB blocks
0000 53
0000 54     V01.08 RN0008      R. Newland      31-Aug-1979
0000 55             31 character symbols (macro names)
0000 56
0000 57     V01.07 RN0005      R. Newland      13-Aug-1979
```

MAC\$MACDEF
V04-000

MACRO DEFINITION PROCESSOR

C 6

16-SEP-1984 02:07:14 VAX/VMS Macro V04-00
5-SEP-1984 01:48:54 [MACRO.SRC]MACDEF.MAR;1

Page 2 (1)

0000 58 :
0000 59 :--
0000 60 ;--

Variable symbol name storage

MA
Ps

SA
MA

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
31
Th
43
16

Ma
--
-\$
-\$
TO
58
Th
MA

0000 62 .SBTTL DECLARATIONS
0000 63 :
0000 64 : INCLUDE FILES:
0000 65 :
0000 66 :
0000 67 :
0000 68 : MACROS:
0000 69 :
0000 70 :
0000 71 \$MAC_SYMBLKDEF ;DEFINE SYMBOL BLOCK OFFSETS
0000 72 \$MAC_INTCODEF ;DEFINE INT. FILE CODES
0000 73 \$MAC_INPBLKDEF ;DEFINE INPUT BLOCK OFFSETS
003C 74 \$MAC_MNBDEF ;DEFINE MNB/MAB OFFSETS
003C 75 \$MAC_GENVALDEF ;DEFINE GENERAL VALUES
003C 76 \$MAC_CREFLAGDEF ;DEFINE CREF CONTROL FLAGS
003C 77 \$MACMSGDEF ;Define message codes
003C 78 \$MAC_CTLFLGDEF ;DEFINE CONTROL FLAGS
003C 79 :
003C 80 :
003C 81 : EQUATED SYMBOLS:
003C 82 :
003C 83 :
00000000 84 .PSECT MAC\$RO_CODE_MAC,NOWRT,GBL,LONG

0000 86 .SBTTL MACRO DEFINE A MACRO
 0000 87
 0000 88 :++
 0000 89 : FUNCTIONAL DESCRIPTION:
 0000 90 :
 0000 91 : THIS ROUTINE IS CALLED WHEN A '.MACRO' DIRECTIVE IS SCANNED.
 0000 92 : THE MACRO FORMAL ARGS ARE SCANNED, AND THE MACRO BODY IS
 0000 93 : SCANNED AND PLACED IN CORE. THE MACRO DEFINITION IS ENTERED
 0000 94 : IN THE USER MACRO HASH TABLE.
 0000 95 :
 0000 96 :--
 0000 97 :
 0000 98 MACRO::: DIRECTIVE = KMACRO
 OE 0005'CF E8 0000 99 BLBS W^LST\$G_MACRODEF+SYMSL VAL.10\$;BRANCH IF LISTING MACRO DEFINITIONS
 FFEA' 30 0013 100 \$INTOUT_LW INT\$ SETLONG,<#0,#MACSGL LIST IT> ;NO--DO NOT LIST
 OB 50 E8 0016 101 10\$: BSBW MAC\$SYM\$CNUP ;SCAN MACRO NAME
 0019 102 BLBS R0,20\$;BRANCH IF NAME FOUND
 5A 0D D0 001E 103 \$MAC_ERR ILLMACNAM ;No--get message code
 FFDC' 31 0021 104 MOVL #CR R10 ;FORCE NEW LINE
 FFD4' 30 0024 105 BRW MAC\$ERRORLN ;ISSUE ERROR AND RETURN
 08 50 E9 002C 106 20\$: MOVAB W^MAC\$AL UMCHSHTB,R3 ;POINT TO USER MACRO HASH TABLE
 52 DD 002F 107 BSBW MAC\$SRC\$MTAB ;LOOK UP THE MACRO NAME
 00BE 30 0031 108 BLBC R0,30\$;BRANCH IF NOT FOUND
 52 8ED0 0034 109 PUSHL R2 ;FOUND--SAVE PTR TO PREVIOUS
 0126 30 0037 110 BSBW FREE_UP_MACRO ;DELETE OLD DEFINITION
 00'8F 9A 003A 111 POPL R2 ;RESTORE PTR TO PREVIOUS
 FFBF' 30 003E 112 30\$: BSBW MACRO_DEFINE ;DEFINE THE MACRO
 FFBC' 30 0041 113 MOVZBL #CRFSR DEF,R5 ;THIS IS MACRO DEFINITION
 2C 5A 91 0044 114 BSBW MAC\$CREF MACRO ;CREF IT IF CREFPING
 03 12 0047 115 BSBW MAC\$SKIPSP ;SKIP SPACES
 FFB4' 30 0049 116 CMPB R10,#^A/.,/ ;STOP ON A COMMA?
 004C 117 BNEQ 40\$;IF NEQ NO
 004C 118 BSBW MAC\$GETCHR ;YES--SKIP IT
 004C 119 :
 004C 120 : SCAN FORMAL ARGUMENTS
 0161 30 004C 121 :
 004F 122 40\$: BSBW MACRO_ARG_SCAN ;SCAN FORMAL ARGUMENTS AND SET UP
 05 A6 0000'CF D0 004F 123 :MAB'S FOR THEM
 SA 0D 9A 0055 124 MOVL W^MACSGL_ARGPTR,MNBSL_TXTP(R6) ;STORE POINTER TO TEXT
 FFAS' 30 0058 125 MOVZBL #CR R10 ;FORCE READING OF NEW LINE
 18 A6 DD 0058 126 BSBW MAC\$GETCHR ;READ FIRST CHARACTER OF BODY
 0000'CF C3 005E 127 PUSHL MNBSL_ARGP(R6) ;STACK THE ARGUMENT POINTER
 50 0000'CF 0062 128 SUBL3 W^MACSGL_BLKPTR,- ;FIGURE SPACE LEFT IN BLOCK
 0000'CF 50 C3 0066 129 SUBL3 W^MACSGL_ARGPTR,RO ;(THIS GIVES SPACE USED)
 0000'CF DD 006E 130 PUSHL RO,#512,=(SP) ;FIGURE SPACE LEFT AND STACK IT
 0000'CF 03 FB 0072 131 CALLS W^MACSGL_ARGPTR ;STACK POINTER TO FREE SPACE IN BLOCK
 0077 132 #3,W^MAC\$BODY_SCAN ;SCAN THE MACRO BODY AND READ
 0077 133 :IT INTO CORE
 0077 134 :
 0077 135 : MACRO IS SCANNED. SEE IF THERE WAS A SYMBOL NAME FOLLOWING THE
 0077 136 : .ENDM DIRECTIVE.
 0077 137 :
 0000'CF D6 0077 138 50\$: INCL W^MACSGL_MCDEF ;COUNT ANOTHER MACRO DEFINITION
 FF82' 30 007B 139 BSBW MAC\$SYM\$CNUP ;IS THERE A SYMBOL?
 2D 50 E9 007E 140 BLBC R0,60\$;IF LBC NO
 50 0000'CF D0 0081 141 MOVL W^MACSGL_MACPTR,RO ;YES--GET PTR TO MNB OF MACRO
 51 04 A0 9A 0086 142 MOVZBL MNBSB_NAME(R0),R1 ;Get offset to symbol count/name

	50	51	C2	008A	143	SUBL2	R1 R0		; and form its address
	51	80	9A	008D	144	MOVZBL	(R0)+ R1		;GET SIZE OF NAME
	52	0000'CF	9E	0090	145	MOVAB	W^MAC\$AB_TMPSYM,R2		;POINT TO SCANNED NAME
	53	82	9A	0095	146	MOVZBL	(R2)+ R3-		;GET ITS SIZE
62	53	00	60	51	2D	CMPCS	R1 (R0),#0,R3,(R2)		;RIGHT .ENDM FOR THIS MACRO?
			OE	13	009E	BEQL	60\$;IF EQL YES
					00A0	SINTOUT	X INT\$ CHKL		;NO--ALIGN SOURCE AND LISTING
					00A6	SMAC_ERR	ENDWRNGMAC		;Get message code
		FF52'	31	00AB	151	BRW	MAC\$ERRORPT		;ISSUE AND RETURN
			05	00AE	152 60\$:	RSB			

	00AF	154	.SBTTL	DELETE	.MDELETE DIRECTIVE
	00AF	155			
	00AF	156	:++		
	00AF	157	: FUNCTIONAL DESCRIPTION:		
	00AF	158	:		
	00AF	159	:	THIS ROUTINE IS CALLED TO PROCESS THE .MDELETE DIRECTIVE. THE	
	00AF	160	:	ARGUMENT TO THE DIRECTIVE IS A LIST OF MACRO NAMES. IF ANY	
	00AF	161	:	OF THE MACROS NAMED ARE DEFINED, THEY ARE UNDEFINED. THE LITERAL	
	00AF	162	:	PRINTED WITH THE LISTING IS THE NUMBER OF MACROS DELETED.	
	00AF	163	:		
	00AF	164	:--		
	00AF	165			
	7E	D4	00AF	166 DELETE::	
	FF4C'	30	00B1	167 CLRL -(SP)	:DIRECTIVE = KMDELETE
	0D 50	E8	00B4	168 BSBW MAC\$SYMSCNUP	:INITIALIZE COUNT OF MACROS DELETED
			169 BLBS R0,20\$:SCAN A MACRO NAME	
			00B7 170 SMAC_ERR DIRSYNX	:BRANCH IF WE FOUND ONE	
	FF41'	30	00BC	171 BSBW MAC\$ERRORPT	:No-directive syntax error
	5A	0D	00BF	172 MOVL #CR,R10	:REPORT ERROR
	22	11	00C2	173 BRB 50\$:FORCE NEW LINE
53	0000'CF	9E	00C4	174 20\$: MOVAB W^MAC\$AL UMCHSHTB,R3	:FINISH PROCESSING
	FF34'	30	00C9	175 BSBW MAC\$SRC\$MTAB	:LOOK UP IN MACRO HASH TABLE
	04 50	E9	00CC	176 BLBC R0,30\$	
	21	10	00CF	177 BSB8 FREE_UP_MACRO	:BRANCH IF NOT FOUND (EASY OUT)
	6E	D6	00D1	178 INCL (SP)	:FOUND--DELETE DEFINITION
	FF2A'	30	00D3	179 30\$: BSBW MAC\$SKIPSP	:COUNT MACRO AS DELETED
	2C	5A	91	180 CMPB R10,#^A/. /	:SKIP SPACES
	06	12	00D9	181 BNEQ 40\$:STOP ON A COMMA?
	FF22'	30	00DB	182 BSBW MAC\$GETCHR	:IF NEQ NO
	FF1F'	30	00DE	183 BSBW MAC\$SKIPSP	:YES--SKIP OVER IT
	0D	5A	91	184 40\$: CMPB R10,#CR	:SKIP SPACES AGAIN
	CB	12	00E4	185 BNEQ 10\$:AT EOL?
	53 8ED0	00E6	186 50\$: POPL R3	:IF NEQ NO--KEEP SCANNING	
			00E9 187 \$INTOUT_LW INT\$_PRIL,R3	:YES--GET COUNT OF MACROS DELETED	
	05	00F1	188 RSB	:PRINT # MACROS DELETED	

H 6

.SBTTL FREE_UP_MACRO DELETE OLD MACRO DEFINITION

00F2 190 .SBTTL FREE_UP_MACRO DELETE OLD MACRO DEFINITION

00F2 191

00F2 192 :++

00F2 193 : FUNCTIONAL DESCRIPTION:

00F2 194 :

00F2 195 :--

00F2 196

00F2 197 FREE_UP_MACRO:

56 DD 00F2 198 PUSHL R6 ;SAVE R6

62 61 DD 00F4 199 MOVL MNBSL_LINK(R1),MNBSL_LINK(R2) ;Unlink MNB from the list

56 51 DD 00F7 200 MOVL R1,R6 ;SAVE POINTER

53 0000'CF D0 00FA 201 MOVL W^MACSGL_INPUTP,R3 ;POINT TO CURRENT INPUT BLOCK

14 A3 51 D1 00FF 202 10\$: CMPL R1_INPSL_RPTCNT(R3) ;REDEFINING THIS MACRO?

07 12 0103 203 BNEQ 20\$;IF NEQ NO

10 14 A3 1F E3 0105 204 BBLS #31,INPSL_RPTCNT(R3),30\$;YES--FLAG TO DELETE THE OLD DEF.

OE 11 010A 205 BRB 30\$;(SAFETY FIRST!)

53 63 D0 010C 206 20\$: MOVL (R3),R3 ;LINK TO NEXT INPUT BLOCK

00000000'8F 53 D1 010F 207 CMPL R3,#MACSGL_PRMINBL ;LOOKING AT PRIMARY INPUT BLOCK?

E7 12 0116 208 BNEQ 10\$;IF NEQ NO

04 10 0118 209 BSBB MACSDEL_MAC_DEF ;DELETE THE MACRO DEFINITION

56 8ED0 011A 210 30\$: POPL R6 ;RESTORE R6

05 011D 211 RSB

011E 212

011E 213 :++

011E 214 : FUNCTIONAL DESCRIPTION:

011E 215 :

011E 216 : THIS ROUTINE DELETES THE MACRO DEFINITION POINTED TO BY R6.

011E 217 : THE MNB IS NOT DEALLOCATED IF CREFRING MACROS.

011E 218 :

011E 219 :--

011E 220

0000'CF D7 011E 221 MACSDEL_MAC_DEF:

05 A6 D4 0122 222 DECC W^MACSGL_MCDEF ;UN-COUNT THE MACRO DEFINITION

15 0000'CF 01 E0 0125 223 CLRL MNBSL_TXTP(R6) ;CLEAR TEXT POINTER

50 04 A6 9A 012B 224 BBS #CRFS\$0 MACROS,W^MACSGL_CRF_FLG,5\$;BRANCH IF CREFRING MACROS

50 56 50 C3 012F 225 MOVZBL MNBSB_NAME(R6),R0 ;Get offset to symbol count/name

56 08 A6 D0 0133 226 SUBL3 R0,R6,R0 ;and get start address of page

FEC6' 30 0137 227 MOVL MNBSL_PAGP(R6),R6 ;POINT TO THE PAGE LIST

0000'CF D7 013A 228 BSBW MACSDEA_1_PAGE ;Deallocate MNB which is always 1 page

50 1B 11 013E 229 DECL W^MACSGL_MCPGRQ ;One less page for macro definitions

50 0B A6 D0 0140 230 BRB 20\$;GO FREE SOME PAGES

0B A6 D4 0144 231 5\$: MOVL MNBSL_PAGP(R6),R0 ;POINT TO POSSIBLE PAGE TO FREE

56 50 D0 0147 232 CLRL MNBSL_PAGP(R6) ;AND PAGE POINTER

13 13 014A 233 MOVL R0,R6 ;ANY PAGES TO FREE?

50 56 D0 014C 234 BEQL 30\$;IF EQL NO--ALL DONE

56 66 D0 014F 235 10\$: MOVL R6,R0 ;GET SET TO FREE CURRENT PAGE

0000'CF 04 A0 C2 0152 236 MOVL MXBSL_LINK(R6),R6 ;Point R6 to possible next block

FEA5' 30 0158 237 SUBL2 MXBSL_PAGES(R0), - ;Subtract size of block from pages

56 D5 015B 238 W^MACSGL_MCPGRQ ;used for macro definitions

ED 12 015D 239 BSBW MACSDEAL_BLOCK ;Deallocate block of memory

05 015F 240 20\$: TSTL R6 ;ANY MORE PAGES?

241 BNEQ 10\$;IF NEQ YES--GO DO THEM

242 RSB

243 30\$: RSB

	0160	245	.SBTTL MACRO_DEFINE CREATE MNB AND SET IT UP		
	0160	246			
	0160	247	:++		
	0160	248	FUNCTIONAL DESCRIPTION:		
	0160	249	:		
	0160	250	THIS ROUTINE ALLOCATES A PAGE FOR A MACRO NAME BLOCK AND		
	0160	251	MACRO ARG BLOCKS. THE MNB IS INITIALIZED AND THE POINTER		
	0160	252	IS RETURNED IN MAC\$GL_MACPTR.		
	0160	253	:		
	0160	254	--		
	0160	255			
	0160	256	MACRO_DEFINE:		
	52	DD	0160	257 PUSHL R2 :SAVE PTR TO PREVIOUS	
	FE9B	30	0162	258 BSBW MAC\$ALL 1 PAGE :ALLOCATE A PAGE FOR THE MNB	
	0000'CF	D6	0165	259 INCL W^MAC\$GE MCPGRQ :COUNT PAGE FOR MACRO DEFINITION	
	0000'CF	50	0169	260 MOVL R0,W^MAC\$GL_BLKPTR :Set pointer to start of block	
	51	0000'CF	9A	016E	261 MOVZBL W^MAC\$AB_TMPSYM,R1 :Get symbol count
	10	51	91	0173	262 CMPB R1,#SYMSR_TWOCOL :Wide listing format required?
	04	04	19	0176	263 BLSS 10\$:No if LSS
	00 6B	2B	E2	0178	264 BBSS #FLGSV_MAC2COL,(R11),10\$; Set flag for wide macro name listing
			017C	265 10\$: INCL R1 :Include the count byte	
				266 MOVC3 R1,W^MAC\$AB_TMPSYM,(R0) :Copy symbol count/name into MNB	
60	0000'CF	51	D6	017C	267 MOVL R3,R6 :Save address of MNB
	56	53	28	017E	268 MOVL R6,W^MAC\$GL_MACPTR :and return address in MACPTR
	0000'CF	56	00	0184	269 POPL R0 :RESTORE PTR TO PREVIOUS
		50	8ED0	0187	270 MOVL MNBSL_LINK(R0),(R3) :Link in new MNB
	83	60	DO	018C	271 MOVL R6,MNBSL_LINK(R0)
	60	56	DO	0192	272 ADDB3 #1,W^MAC\$AB_TMPSYM,(R3) :Insert name offset
83	0000'CF	01	81	0195	273 CLRL (R3)+ :CLEAR TXTP
	83	D4	0198	274 CLRW (R3)+ :CLEAR FLAGS	
	83	84	019D	275 CLRL (R3)+ :CLEAR PAGP	
	83	D4	019F	276 MOVL #1,(R3)+ :INITIALIZE PAGC (PAGE COUNT)	
83	01	DO	01A1	277 CLRL (R3)+ :CLEAR CRSYM	
	83	D4	01A4	278 CLRB (R3)+ :CLEAR ARG COUNT	
	83	94	01A6	279 CLRL (R3)+ :CLEAR ARG POINTER	
	83	D4	01A8	280 MOVL R3,W^MAC\$GL_ARGPTR :STORE POINTER FOR MAB ALLOCATION	
	0000'CF	53	DO	01AA	281 RSB
		05	01AF	282	

01B0 284 .SBTTL MACRO_ARG_SCAN SCAN FORMAL MACRO ARGUMENTS
 01B0 285
 01B0 286 :++
 01B0 287 : FUNCTIONAL DESCRIPTION:
 01B0 288 :
 01B0 289 : THIS ROUTINE SCANS THE FORMAL MACRO ARGUMENTS AND STORES
 01B0 290 : THEM IN MACRO ARGUMENT BLOCKS (MAB'S) THAT ARE LINKED TO
 01B0 291 : THE MACRO NAME BLOCK.
 01B0 292 :
 01B0 293 :--
 01B0 294 :
 01B0 295 MACRO_ARG_SCAN:
 58 1180 8F 88 01B0 296 POSHR #^MCR7,R8,R12>
 50 0000'CF D0 01B4 297 MOVL W^MACSGL ARGPTR,R8
 57 04 A6 9A 01B9 298 MOVZBL MNBSB NAME(R6),R0
 57 58 56 C3 01BD 299 SUBL3 R6,R8,R7
 57 50 C0 01C1 300 ADDL2 R0,R7
 57 FE31' C3 01C4 301 SUBL3 R7,#512,R7
 0D 5A 91 01CC 302 BSBW MAC\$SKIPSP
 03 12 01D2 303 CMPB R10,#CR
 0150 31 01D4 304 BNEQ 10\$
 01D7 305 BRW ARG_SCAN_EXIT
 01D7 306 10\$:
 01D7 307
 01D7 308 ARG_SCAN_LOOP:
 3F FE26' 30 01D7 309 BSBW MAC\$SKIPSP
 5A 91 01DA 310 CMPB R10,#^A/?/
 15 12 01DD 311 BNEQ 30\$
 1F 17 A6 91 01DF 312 CMPB MNBSB_ARGCT(R6),#31
 OF 1A 01E3 313 BGTRU 30\$
 00 13 A6 9A 01E5 314 MOVZBL MNBSB_ARGCT(R6),R0
 50 50 E3 01E9 315 BBCS R0,MNBSL CRSYMF(R6),20\$
 FEOF' 30 C1EE 316 20\$: BSBW MAC\$GETCAR
 FEOC' 30 01F1 317 BSBW MAC\$SKIPSP
 02 E0 01F4 318 30\$: BBS #CHR\$V SYM CHR,-
 2F 0000'CA 01F6 319 W^MACSAB CMSK_TAB(R10),80\$
 FDFE' 30 01FA 320 \$MAC_ERR ILLMACARGN
 0202 321 BSBW MACERRORLN
 0202 322 :
 0202 323 : LOOP LOOKING FOR END OF ARGUMENT OR END OF LINE
 0202 324 :
 2C FDFB' 30 0202 325 40\$: BSBW MAC\$GETCHR
 5A 91 0205 326 CMPB R10,#^A/./
 11 13 0208 327 BEQL 50\$
 09 5A 91 020A 328 CMPB R10,#TAB
 OF 13 020D 329 BEQL 60\$
 20 5A 91 020F 330 CMPB R10,#^A/ /
 0A 13 0212 331 BEQL 60\$
 0D 5A 91 0214 332 CMPB R10,#CR
 0D 13 0217 333 BEQL 70\$
 E7 11 0219 334 BRB 40\$
 021B 335 :
 021B 336 : FOUND A COMMA
 021B 337 :
 FDE2' 30 021B 338 50\$: BSBW MAC\$GETCHR
 FDDF' 30 021E 339 60\$: BSBW MAC\$SKIPSP
 0D 5A 91 0221 340 CMPB R10,#CR
 : SAVE REGISTERS
 : POINT TO FREE SPACE IN BLOCK
 : Get bytes used to store symbol name
 : Figure bytes used for fixed-sized MNB
 : Add fixed and variable parts
 : REALLY GET SPACE LEFT NOW
 : SKIP SPACES
 : SKIP TO END OF LINE?
 : IF NEQ NO
 : YES--NO FORMAL ARGS HERE
 : SKIP LEADING SPACES
 : CREATED SYMBOL?
 : IF NEQ NO
 : ONLY FIRST 32 SYMBOLS CAN BE CREATED
 : IF GTR THEN LOSE
 : GET ARGUMENT NUMBER - 1
 : SET CREATED SYMBOL FLAG FOR THIS ARG
 : SKIP THE QUESTION MARK
 : SKIP LEADING SPACES
 : BRANCH IF LEGAL CHAR FOR ARG
 : No--get error code
 : ISSUE ERROR TO PASS 2
 : GET NEXT CHARACTER
 : START OF NEXT ARG?
 : IF EQL YES
 : TAB?
 : IF EQL YES
 : SPACE?
 : IF EQL YES
 : NO--END OF LINE?
 : IF EQL YES
 : NO--KEEP SKIPPING
 : SKIP THE COMMA
 : SKIP SPACES
 : DID WE SKIP TO END OF LINE?

B1	12	0224	341	BNEQ	ARG SCAN_LOOP	;IF NEQ NO--CONTINUE SCANNING ARGS		
00F9	31	0226	342	70\$: BRW	150\$;QUIT ON EOL		
		0229	343					
		0229	344	: CHARACTER CAN LEGALLY START A MACRO ARGUMENT				
		0229	345					
FDD4'	30	0229	346	80\$: BSBW	MAC\$SYMSCNUP	;GET SYMBOL NAME		
56	DD	022C	347	PUSHL	R6	;SAVE PTR TO MNB		
56	18	A6	022E	348	MOVAB	MNBSL_ARGP(R6),R6	;POINT TO ARGUMENT LIST	
55	56	56	0232	349	MOVL	R6,R5	;MAKE A COPY	
FDC8'	30	0235	350	BSBW	MAC\$SRC_LIST	;LOOK IT UP		
56	8ED0	0238	351	POPL	R6			
50	DD	023B	352	PUSHL	RO			
0000'CF	51	DO	023D	353	MOVL	R1,W^MAC\$GL_ARGPTR		
53	06	A1	0242	354	MOVAL	MABSW_DVLEN(R1),R3		
63	50	E8	0246	355	BLBS	RO,100\$		
50	53	58	0249	356	MOVL	R8,R3		
0000'CF	9A	024C	357	MOVZBL	W^MAC\$AB_TMPSYM,RO			
50	D6	0251	358	INCL	RO			
57	50	C2	0253	359	SUBL2	RO,R7		
57	OC	C2	0256	360	SUBL2	#MABSK_BLKSIZ,R7		
25	14	0259	361	BGTR	90\$			
FDA2'	30	025B	362	BSBW	MAC\$ALL 1 PAGE			
0000'CF	D6	025E	363	INCL	W^MAC\$GE_MCPGRQ			
58	50	DO	0262	364	MOVL	RO,R8		
0000'CF	50	DO	0265	365	MOVL	RO,W^MAC\$GL_BLKPTR		
			026A	366	ASSUME	MXBSDL_LINK EQ 0		
			026A	367	ASSUME	MXBSDL_PAGES EQ MXBSDL_LINK+4		
88	0B	A6	DO	026A	368	MOVL	MNBSL_PAGP(R6),(R8)+	
0B	A6	50	DO	026E	369	MOVL	RO,MNBSL_PAGP(R6)	
88	01	DO	0272	370	MOVL	#1,(R8)+		
OF	A6	D6	0275	371	INCL	MNBSL_PAGC(R6)		
57	01EC	8F	3C	0278	372	MOVZWL	#<512-MXBSDL_BLKSIZ-MABSK_BLKSIZ>,R7 : Set block count	
53	58	DO	027D	373	MOVL	R8,R3		
			0280	374	90\$:			
50	0000'CF	52	DD	0280	375	PUSHL	R2	
50	9A	0282	376	MOVZBL	W^MAC\$AB_TMPSYM,RO			
50	D6	0287	377	INCL	RO			
0000'CF	50	28	0289	378	MOVC3	RO,W^MAC\$AB_TMPSYM,(R3)		
0000'CF	53	DO	028F	379	MOVL	R3,W^MAC\$GL_ARGPTR		
52	8ED0	0294	380	POPL	R2			
83	62	DO	0297	381	MOVL	MABSL_LINK(R2),(R3)+		
62	0000'CF	DO	029A	382	MOVL	W^MAC\$GL_ARGPTR,MABSL_LINK(R2) ; Point previous link to new MAB		
0000'CF	01	81	029F	383	ADDB3	#1,W^MAC\$AB_TMPSYM,(R3)+		
83	17	A6	96	02A5	384	INC8	MNBSB_ARGCT(R6)	
83	17	A6	90	02A8	385	MOVB	MNBSB_ARGCT(R6),(R3)+	
5C	53	DO	02AC	386	100\$:	MOVL	R3,R12	
83	B4	02AF	387	CLRW	(R3)+			
83	D4	02B1	388	CLRL	(R3)+			
8E	D5	02B3	389	TSTL	(SP)+			
03	12	02B5	390	BNEQ	110\$			
58	53	DO	02B7	391	MOVL	R3,R8		
FD43'	30	02BA	392	110\$:	BSBW	MAC\$SKIPSP		
3D	5A	91	02BD	393	CMPB	R10,#^A/=		
50	12	02C0	394	BNEQ	130\$			
			02C2	395				
			02C2	396	:	SCAN DEFAULT ARGUMENT		
			02C2	397	:			

	FD3B'	30	02C2	398	BSBW	MAC\$GETCHR	:GET FIRST CHARACTER OF ARGUMENT	
	FD38'	30	02C5	399	BSBW	MAC\$MAC_ARG_SCN	:SCAN DEFAULT ARG	
	50	D5	02C8	400	TSTL	R0	:WAS IT NULL?	
	46	13	02CA	401	BEQL	130\$:IF EQL YES	
	8C	50	80	02CC	402	MOVW	:STORE DEFAULT VALUE LENGTH	
	57	50	C2	02CF	403	SUBL2	RO,(R12)+	
	32	14	02D2	404	BGTR	RO,R7	:IS THERE ROOM FOR DEFAULT STRING?	
	50	DD	02D4	405	PUSHL	120\$:IF GTR YES	
S1	50	08	C1	02D6	406	ADDL3	#MXBSK_BLKSIZ,R0,R1	
	FD23'	30	02DA	407	BSBW	MAC\$ALLOC_BLOCK		
0000'CF	51	CO	02DD	408	ADDL2	R1,W^MAC\$GL_MCPGRQ		
	58	50	DO	02E2	409	MOVL	R0,R8	
0000'CF	50	DO	02E5	410	MOVL	R0,W^MAC\$GL_BLKPTR		
	88	OB	A6	DO	02EA	411	MOVL	MNB\$L_PAGP(R6),(R8)+
	OB	A6	50	DO	02EE	412	MOVL	R0,MNB\$L_PAGP(R6)
	88	51	DO	02F2	413	MOVL	R1,(R8)+	
	OF	A6	51	CO	02F5	414	ADDL2	R1,MNB\$L_PAGC(R6)
	57	51	09	78	02F9	415	ASHL	#9,R1,R7
	57	08	C2	02FD	416	SUBL2	#MXBSK_BLKSIZ,R7	
	50	8ED0	0300	417	POPL	RO		
	57	50	C2	0303	418	SUBL2	RO,R7	
	8C	58	DO	0306	419	120\$:	MOVL	R8,(R12)+
	0000'CF	50	28	0309	420	MOVC3	RO,W^MAC\$AB_TMPBUF,(R8)	
	58	53	DO	030F	421	MOVL	R3,R8	
	2C	5A	91	0312	422	130\$:	CMPB	R10,#^A//.
	03	12	0315	423	BNEQ	140\$		
	FCE6'	30	0317	424	BSBW	MAC\$GETCHR		
	0D	5A	91	031A	425	140\$:	CMPB	R10,#CR
	03	13	031D	426	BEQL	150\$		
	FEB5	31	031F	427	BRW	ARG_SCAN_LOOP		
0000'CF	58	DO	0322	428	150\$:	MOVL	R8,W^MAC\$GL_ARGPTR	
	0327	429	ARG_SCAN_EXIT:				:NO--CONTINUE SCANNING ARGS	
	1180 8F	BA	0327	430	POPR	#^M<R7,R8,R12>	:STORE UPDATED ARG POINTER	
	05	032B	431				:RESTORE REGISTERS	
	032C	432						
	032C	433						
	032C	434	.END					

\$COUNT	= 0000003B		FLG\$M_INSERT	= 00000100	FLG\$V_MACLTB	= 0000001B
ARG\$K_SIZE	= 000003E8		FLG\$M_IRPC	= 20000000	FLG\$V_MACTXT	= 00000010
ARG_SCAN_EXIT	= 00000327 R	03	FLG\$M_LEXOP	= 00000002	FLG\$V_MEBLST	= 0000000C
ARG_SCAN_LOOP	= 000001D7 R	03	FLG\$M_LSTXST	= 00000200	FLG\$V_MOREARG	= 0000002D
AUD\$K_SIZE	= 00000010		FLG\$M_MAC2COL	= 00000800	FLG\$V_MOREINP	= 00000023
BLNK	= 00000020		FLG\$M_MACL	= 00000800	FLG\$V_NEWPND	= 0000000A
CHR\$M_COMMACR	= 00000020		FLG\$M_MACLTB	= 08000000	FLG\$V_NOREF	= 00000018
CHR\$M_ILL_CCR	= 00000040		FLG\$M_MACTXT	= 00010000	FLG\$V_NTYPEDPC	= 00000025
CHR\$M_NUM_BER	= 00000010		FLG\$M_MEBLST	= 00001000	FLG\$V_NULCHR	= 00000032
CHR\$M_SPA_MSK	= 00000001		FLG\$M_MOREARG	= 00002000	FLG\$V_OBJXST	= 00000015
CHR\$M_SYM_CH1	= 00000008		FLG\$M_MOREINP	= 00000008	FLG\$V_OPNDCHK	= 00000028
CHR\$M_SYM_CHR	= 00000004		FLG\$M_NEWPND	= 00000400	FLG\$V_OPRND	= 0000000D
CHR\$M_SYM_DLM	= 00000002		FLG\$M_NOREF	= 01000000	FLG\$V_OPTVFLIDX	= 0000002C
CHR\$V_COMMACR	= 00000005		FLG\$M_NTYPEDPC	= 00000020	FLG\$V_ORDLST	= 00000011
CHR\$V_CVTLWC	= 00000061		FLG\$M_NULCHR	= 00040000	FLG\$V_P2	= 0000000E
CHR\$V_ILL_CHR	= 00000006		FLG\$M_OBJXST	= 00200000	FLG\$V_RPTIRP	= 0000001C
CHR\$V_NOCT	= 0000007F		FLG\$M_OPNDCHK	= 00000100	FLG\$V_SEQFIL	= 00000019
CHR\$V_NUM_BER	= 00000004		FLG\$M_OPRND	= 00002000	FLG\$V_SKAN	= 0000000F
CHR\$V_SPA_MSK	= 00000000		FLG\$M_OPTVFLIDX	= 00001000	FLG\$V_SPECOP	= 00000022
CHR\$V_SYM_CH1	= 00000003		FLG\$M_ORDLST	= 00020000	FLG\$V_SPLALL	= 0000001A
CHR\$V_SYM_CHR	= 00000002		FLG\$M_P2	= 00004000	FLG\$V_STOIMF	= 00000012
CHR\$V_SYM_DLM	= 00000001		FLG\$M_RPTIRP	= 10000000	FLG\$V_SYM2COL	= 0000002A
CNT	= 00000001		FLG\$M_SEQFIL	= 02000000	FLG\$V_TOCFLG	= 00000013
CR	= 0000000D		FLG\$M_SKAN	= 00008000	FLG\$V_UPAFLG	= 00000024
CRFSK_DEF	*****	X 03	FLG\$M_SPECOP	= 00000004	FLG\$V_UPDFIL	= 00000027
CRFSM_DEFAULT	= 00000012		FLG\$M_SPLALL	= 04000000	FLG\$V_UPMARG	= 00000026
CRFSM_DIR	= 00000001		FLG\$M_STOIMF	= 00040000	FLG\$V_XCRF	= 0000001F
CRFSM_MACROS	= 00000002		FLG\$M_SYM2COL	= 00000400	FREE_UP_MACRO	= 000000F2 R 03
CRFSM_OPCODES	= 00000004		FLG\$M_TOCFLG	= 00080000	HASHSZ	= 0000007F
CRFSM_REGISTERS	= 00000008		FLG\$M_UPAFLG	= 00000010	HYPHEN	= 0000002D
CRFSM_SYMBOLS	= 00000010		FLG\$M_UPDFIL	= 00000080	INPSB_ARGCT	= 0000001C
CRFSV_DIR	= 00000000		FLG\$M_UPMARG	= 00000040	INPSK_BLKSIZ	= 00000021
CRFSV_MACROS	= 00000001		FLG\$M_XCRF	= 80000000	INPSK_BUFSIZ	= 000003E8
CRFSV_OPCODES	= 00000002		FLG\$V_ALLCHR	= 00000000	INPSK_IRPSIZ	= 0000003C
CRFSV_REGISTERS	= 00000003		FLG\$V_BOL	= 00000001	INPSL_ARGS	= 0000001D
CRFSV_SYMBOLS	= 00000004		FLG\$V_CHKLPND	= 00000014	INPSL_GETL	= 00000008
DELETE	= 000000AF RG	03	FLG\$V_COMPEXPR	= 00000002	INPSL_IFLVL	= 0000000C
ERR	= 00000000		FLG\$V_CONT	= 00000003	INPSL_IFVAL	= 00000010
FF	= 0000000C		FLG\$V_CRF	= 0000001E	INPSL_LINK	= 00000000
FLG\$M_ALLCHR	= 00000001		FLG\$V_CRSEEN	= 00000020	INPSL_NXTL	= 00000004
FLG\$M_BOL	= 00000002		FLG\$V_DATRPT	= 00000004	INPSL_PAGP	= 00000018
FLG\$M_CHKLPND	= 00100000		FLG\$V_DBGOUT	= 0000002E	INPSL_RPTCNT	= 00000014
FLG\$M_COMPEXPR	= 00000004		FLG\$V_DLIMSTR	= 0000002F	INTSK_BUFSIZ	= 000013F4
FLG\$M_CONT	= 00000008		FLG\$V_ENDMCH	= 00000005	INTSK_BUFWRN	= 00001390
FLG\$M_CRF	= 40000000		FLG\$V_EVALEXPR	= 00000006	INT\$_ADD	= 00000001
FLG\$M_CRSEEN	= 00000001		FLG\$V_EXPOPT	= 00000007	INT\$_AND	= 00000002
FLG\$M_DATRPT	= 00000010		FLG\$V_EXTERR	= 00000030	INT\$_ASH	= 00000003
FLG\$M_DBGOUT	= 00004000		FLG\$V_EXTWRN	= 00000031	INT\$_ASN	= 0000000C
FLG\$M_DLIMSTR	= 00008000		FLG\$V_FIRSTTLN	= 00000029	INT\$_AUGPC	= 0000000D
FLG\$M_ENDMCH	= 00000020		FLG\$V_IFSTAT	= 00000017	INT\$_BDST	= 0000000E
FLG\$M_EVALEXPR	= 00000040		FLG\$V_IIF	= 00000016	INT\$_CHKL	= 0000000F
FLG\$M_EXPOPT	= 00000080		FLG\$V_INSERT	= 00000008	INT\$_DIV	= 00000004
FLG\$M_EXTERR	= 00010000		FLG\$V_IRPC	= 0000001D	INT\$_END	= 00000010
FLG\$M_EXTWRN	= 00020000		FLG\$V_LEXOP	= 00000021	INT\$_EPT	= 00000011
FLG\$M_FIRSTTLN	= 00000200		FLG\$V_LSTXST	= 00000009	INT\$_ERR	= 00000012
FLG\$M_IFSTAT	= 00800000		FLG\$V_MAC2COL	= 0000002B	INT\$_ETX	= 00000013
FLG\$M_IIF	= 00400000		FLG\$V_MACL	= 0000000B	INT\$_FNEWL	= 00000014

INT\$_ILG	= 00000000	MAC\$AB_TMPBUF	*****	X	03	PSC\$B_SEG	0000000C
INT\$_INFO	= 0000003A	MAC\$AB_TMPSYM	*****	X	03	PSC\$B_UNUSED	00000008
INT\$_LGLAB	= 000C0015	MAC\$ALC_1_PAGE	*****	X	03	PSC\$K_BLKSIZ	00000013
INT\$_MACL	= 00000016	MAC\$ALL_BLOCK	*****	X	03	PSC\$K_NO_OPTNS	= 0000000A
INT\$_MUL	= 00000005	MAC\$AL_DMCHSHTB	*****	X	03	PSC\$L_CURLOC	0000000F
INT\$_NEG	= 00000006	MAC\$BODY_SCAN	*****	X	03	PSC\$L_LINK	00000000
INT\$_NEWL	= 00000017	MAC\$CREF_MACRO	*****	X	03	PSC\$L_MAXLTH	00000005
INT\$_NEWP	= 00000018	MAC\$DEAL_BLOCK	*****	X	03	PSC\$M_ABS	= FFFFFFF7
INT\$_NOT	= 00000007	MAC\$DEA_T_PAGE	*****	X	03	PSC\$M_ALIGNFLG	= 00004000
INT\$_OP	= 00000019	MAC\$DEL_MAC_DEF	0000011E	RG	03	PSC\$M_ALLOPTNS	= 00003FF
INT\$_OR	= 00000008	MAC\$ERRRLN	*****	X	03	PSC\$M_BYTE	= 00004000
INT\$_PRIL	= 0000001A	MAC\$ERRORPT	*****	X	03	PSC\$M_CON	= FFFFFFFB
INT\$_PRT	= 0000001B	MAC\$GETCHR	*****	X	03	PSC\$M_DEFAULT	= 00001C8
INT\$_PSECT	= 0000001C	MAC\$GL_ARGPTR	*****	X	03	PSC\$M_EXE	= 000000C0
INT\$_REDEF	= 0000001D	MAC\$GL_BLKPTR	*****	X	03	PSC\$M_GBL	= 00000010
INT\$_REF	= 0000001E	MAC\$GL_CRF_FLG	*****	X	03	PSC\$M_LCL	= FFFFFFEF
INT\$_REST	= 0009001F	MAC\$GL_INPUTP	*****	X	03	PSC\$M_LIB	= 00000002
INT\$_SAME	= 00000009	MAC\$GL_LIST_IT	*****	X	03	PSC\$M_LONG	= 00004800
INT\$_SAVE	= 00000020	MAC\$GL_MACPTR	*****	X	03	PSC\$M_NOEXE	= FFFFFFFBF
INT\$_SBTL	= 00000021	MAC\$GL_MCDEF	*****	X	03	PSC\$M_NOPIC	= FFFFFFFE
INT\$_SETFLAG	= 00000022	MAC\$GL_MCPGRQ	*****	X	03	PSC\$M_NORD	= FFFFFF7F
INT\$_SETLONG	= 00000023	MAC\$GL_PRMINBL	*****	X	03	PSC\$M_NOSHR	= FFFFFFD
INT\$_SPIC	= 00000024	MAC\$INTOUT_1_LW	*****	X	03	PSC\$M_NOVEC	= FFFFFDFF
INT\$_SPID	= 00000025	MAC\$INTOUT_2_LW	*****	X	03	PSC\$M_NOWRT	= FFFFFEFF
INT\$_STIB	= 00000026	MAC\$INTOUT_X	*****	X	03	PSC\$M_OVR	= 00000004
INT\$_STIL	= 00000028	MAC\$MAC_ARG_SCN	*****	X	03	PSC\$M_PAGE	= 00006400
INT\$_STIW	= 00000027	MAC\$SKIPSP	*****	X	03	PSC\$M_PIC	= 00000001
INT\$_STKEPT	= 00000029	MAC\$SRC_SYMTAB	*****	X	03	PSC\$M_QUAD	= 00004C00
INT\$_STKG	= 0000002A	MAC\$SRC_LIST	*****	X	03	PSC\$M_RD	= 00000080
INT\$_STKL	= 0000002B	MAC\$SYM5CNUP	*****	X	03	PSC\$M_REL	= 00000008
INT\$_STKPC	= 0000002C	MAC\$DIRSYNX	= 007D906A			PSC\$M_SHR	= 00000020
INT\$_STKS	= 0000002D	MAC\$ENDWRNGMAC	= 007D907A			PSC\$M_USR	= FFFFFFFD
INT\$_STOB	= 00000034	MAC\$ILLMACARGN	= 007D90EA			PSC\$M_VEC	= 0000200
INT\$_STOL	= 0000002E	MAC\$ILLMACNAM	= 007D90F2			PSC\$M_WORD	= 00004400
INT\$_STOW	= 00000035	MACRO	00000000	RG	03	PSC\$M_WRT	= 0000180
INT\$_STRB	= 0000002F	MACRO_ARG_SCAN	00000180	R	03	PSC\$S_ALIGNMENT	= 00000004
INT\$_STRL	= 00000031	MACRO_DEFINE	00000160	R	03	PSC\$V_ALIGNMENT	= 0000000E
INT\$_STRSB	= 00000032	MAC\$SUBSYS	= 0000007D			PSC\$V_EXE	= 00000006
INT\$_STRSW	= 00000033	MNB\$B_ARGCT	00000017			PSC\$V_GBL	= 00000004
INT\$_STRW	= 00000030	MNB\$B_NAME	00000004			PSC\$V_LIB	= 00000001
INT\$_STSBB	= 00000036	MNB\$K_BLKSIZ	0000001C			PSC\$V_OVR	= 00000002
INT\$_STSW	= 00000037	MNB\$L_ARGP	00000018			PSC\$V_PIC	= 00000000
INT\$_SUB	= 0000000A	MNB\$L_CRSYMF	00000013			PSC\$V_RD	= 00000007
INT\$_SUME	= 00000039	MNB\$L_LINK	00000000			PSC\$V_REL	= 00000003
INT\$_WRH	= 00000038	MNB\$L_PAGC	0000000F			PSC\$V_SHR	= 00000005
INT\$_XOR	= 0000000B	MNB\$L_PAGP	0000000B			PSC\$V_VEC	= 00000009
LST\$G_MACRODEF	*****	MNB\$L_TXTP	00000005			PSC\$V_WRT	= 00000008
LST\$K_BUFSIZ	= 00000086	MNB\$W_FLAG	00000009			PSC\$W_FLAG	= 00000009
LST\$K_L_P_PAGE	= 0000003C	MXB\$K_BLKSIZ	00000008			PSC\$W_OPTIONS	= 00000000
LST\$K_TITLE_SIZE	= 00000028	MXB\$L_LINK	00000000			RDX\$V_BINARY	= 00000000
MAB\$B_ARGNO	00000005	MXB\$L_PAGES	00000004			RDX\$V_DECIMAL	= 00000002
MAB\$B_NAME	00000004	OBJ\$K_BUFSIZ	= 00000200			RDX\$V_DOUBLE	= 00000005
MAB\$K_BLKSIZ	0000000C	OPF\$M_LASTOPR	= 00002000			RDX\$V_FLOAT	= 00000004
MAB\$L_DV PTR	00000008	OPF\$M_OPTEXP	= 00001000			RDX\$V_GFLOAT	= 00000006
MAB\$L_LINK	00000000	OPF\$V_LASTOPR	= 0000000D			RDX\$V_HEX	= 00000003
MAB\$W_DVLEN	00000006	OPF\$V_OPTEXP	= 0000000C			RDX\$V_HFLOAT	= 00000007
MAC\$AB_CMSK_TAB	*****	PSC\$B_NAME	00000004				

```

RDX$V_OCTAL      = 00000001
REGS_PC          = 0000000F
SEMI              = 0000003B
STB$K_PG_MISS   = 0000000A
SYM$B_NAME        = 00000004
SYM$B_SEG         = 0000000C
SYM$B_TOKEN       = 0000000B
SYM$K_BLKSIZ     = 0000000D
SYM$K_MAXLEN     = 0000001F
SYM$K_TWOCOL     = 00000010
SYM$L_LINK        = 00000000
SYM$L_VAL         = 00000005
SYM$M_ABS         = 00000010
SYM$M ASN        = 00000100
SYM$M_CRF         = 00002000
SYM$M_DEBUG       = 00000020
SYM$M_DEF         = 00000001
SYM$M_DELMAC     = 00000200
SYM$M_EPT         = 00000200
SYM$M_EXTRN       = 00000008
SYM$M_GLOBL       = 00000004
SYM$M_LOCAL       = 00000040
SYM$M_ODBG        = 00000400
SYM$M_REF         = 00000080
SYM$M_RELSECT     = 00000800
SYM$M_SUPR        = 00004000
SYM$M_WEAK        = 00000002
SYM$V_XCRF        = 0001000
SYM$V_ABS          = 00000004
SYM$V ASN         = 00000008
SYM$V_CRF         = 0000000D
SYM$V_DEBUG        = 00000005
SYM$V_DEF          = 00000000
SYM$V_DELMAC      = 00000009
SYM$V_EPT          = 00000009
SYM$V_EXTRN        = 00000003
SYM$V_GLOBL        = 00000002
SYM$V_LOCAL        = 00000006
SYM$V_ODBG        = 0000000A
SYM$V_REF          = 00000007
SYM$V_RELSECT      = 0000000B
SYM$V_SUPR         = 0000000E
SYM$V_WEAK         = 00000001
SYM$V_XCRF        = 0000000C
SYM$W_FLAG         = 00000009
TAB               = 00000009
X1                = 00000033
X2                = 00080000

```

+-----+
! Psect synopsis !
+-----+

PSECT name

```
-----  
: ABS :  
. BLANK :
```

Allocation

```
-----  
00000000 ( 0.)  
00000000 ( 0.)
```

PSECT No.

```
-----  
00 ( 0.)  
01 ( 1.)
```

Attributes

```
-----  
NOPIC USR  
NOPIC USR
```

CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

MAC\$MACDEF
Psect synopsis

SABSS
MAC\$RO_CODE_MAC

MACRO DEFINITION PROCESSOR

C 7

16-SEP-1984 02:07:14 VAX/VMS Macro V04-00
5-SEP-1984 01:48:54 [MACRO.SRC]MACDEF.MAR;1

Page 15
(7)

MA
VO

0000003C (60.) 02 (2.) NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
0000032C (812.) 03 (3.) NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:00.82
Command processing	117	00:00:00.37	00:00:04.41
Pass 1	214	00:00:03.45	00:00:13.78
Symbol table sort	0	00:00:00.48	00:00:01.66
Pass 2	95	00:00:00.84	00:00:03.45
Symbol table output	32	00:00:00.19	00:00:01.72
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	491	00:00:05.41	00:00:25.87

The working set limit was 1350 pages.

31798 bytes (63 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 466 non-local and 34 local symbols.

434 source lines were read in Pass 1, producing 19 object records in Pass 2.

16 pages of virtual memory were used to define 15 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	13
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	17

586 GETS were required to define 17 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:MACDEF/OBJ=OBJ\$:MACDEF MSRC\$:MACDEF/UPDATE=(ENHS:MACDEF)+LIB\$:MACRO/LIB

0226 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

